



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE

United States Patent and Trademark Office

Address: COMMISSIONER FOR PATENTS

P.O. Box 1450

Alexandria, Virginia 22313-1450

www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/720,960	11/24/2003	Richard D. Dettinger	ROC920030277US1	5201
46797	7590	08/02/2010		
IBM CORPORATION, INTELLECTUAL PROPERTY LAW DEPT 917, BLDG. 006-1 3605 HIGHWAY 52 NORTH ROCHESTER, MN 55901-7829			EXAMINER DWIVEDI, MAHESH H	
			ART UNIT	PAPER NUMBER
			2168	
			MAIL DATE	DELIVERY MODE
			08/02/2010	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Office Action Summary

Application No.

10/720,960

Applicant(s)

DETTINGER ET AL.

Examiner

MAHESH H. DWIVEDI

Art Unit

2168

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --
Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

- 1) ☒ Responsive to communication(s) filed on 27 July 2010.
- 2a) ☒ This action is **FINAL**. 2b) ☐ This action is non-final.
- 3) ☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

- 4) ☒ Claim(s) 10, 14-16, 20, 21 and 27-32 is/are pending in the application.
- 4a) Of the above claim(s) _____ is/are withdrawn from consideration.
- 5) ☐ Claim(s) _____ is/are allowed.
- 6) ☒ Claim(s) 10, 14-16, 20, 21 and 27-32 is/are rejected.
- 7) ☐ Claim(s) _____ is/are objected to.
- 8) ☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

- 9) ☐ The specification is objected to by the Examiner.
- 10) ☒ The drawing(s) filed on 24 November 2003 is/are: a) ☒ accepted or b) ☐ objected to by the Examiner.
Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).
Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).
- 11) ☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

- 12) ☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).
- a) ☐ All b) ☐ Some * c) ☐ None of:
1. ☐ Certified copies of the priority documents have been received.
 2. ☐ Certified copies of the priority documents have been received in Application No. _____.
 3. ☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

- 1) ☒ Notice of References Cited (PTO-892)
- 2) ☐ Notice of Draftperson's Patent Drawing Review (PTO-948)
- 3) ☐ Information Disclosure Statement(s) (PTO/SB/08)
Paper No(s)/Mail Date _____
- 4) ☐ Interview Summary (PTO-413)
Paper No(s)/Mail Date _____
- 5) ☐ Notice of Informal Patent Application
- 6) ☐ Other: _____

DETAILED ACTION

Remarks

1. Receipt of Applicant's Amendment, filed on 07/27/2010, is acknowledged. The amendment includes the cancellation of Claims 1-9, 11-13, 17-19, and 22-26.

Claim Rejections - 35 USC § 103

2. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

(a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains.

Patentability shall not be negated by the manner in which the invention was made.

3. This application currently names joint inventors. In considering patentability of the claims under 35 U.S.C. 103(a), the examiner presumes that the subject matter of the various claims was commonly owned at the time any inventions covered therein were made absent any evidence to the contrary. Applicant is advised of the obligation under 37 CFR 1.56 to point out the inventor and invention dates of each claim that was not commonly owned at the time a later invention was made in order for the examiner to consider the applicability of 35 U.S.C. 103(c) and potential 35 U.S.C. 102(e), (f) or (g) prior art under 35 U.S.C. 103(a).
4. Claims 10, 14-16, 20-21, and 27-32 are rejected under 35 U.S.C. 103(a) as being unpatentable over **Win et al.** (U.S. Patent 6,453,353) in view of **Faybishenko et al.** (U.S. PGPUB 2003/0158839), and further in view of **Chang et al.** (U.S. Patent 6,968,509)
5. Regarding claim 10, **Win** teaches a method comprising:
 - A) assigning metadata requirements to functional modules that operate on data stored in, or functional modules that generate results that are stored in, a database (Abstract, Column 5, lines 44-46, Column 6, lines 10-16, lines 41-65);
 - B) wherein the assigned metadata requirements specify conditions required for successful execution of the functional module (Abstract, Column 5, lines 44-46, Column 6, lines 10-16, lines 41-65);

- C) wherein at least one condition defines at least one user role required for successful; execution of the functional module (Abstract, Column 5, lines 44-46, Column 6, lines 10-16, lines 41-65);
- H) obtaining a list of functional modules that are accessible from within the application used during the query session (Abstract, Column 6, lines 10-16, lines 41-65);
- I) identifying a limited subset of the functional modules in the list that will successfully execute, by comparing the collected runtime metadata with the assigned metadata requirements (Abstract, Column 6, lines 10-16, lines 41-65); and
- J) providing an interface presenting the user with the identified limited subset of functional modules that will successfully execute (Abstract, Column 6, lines 10-16, lines 41-65).

The examiner notes that **Win** teaches **“assigning metadata requirements to functional modules that operate on data stored in, or functional modules that generate results that are stored in, a database”** as “Roles determine what resources a User can access. Further, each role may require a set of information that is available in resources” (Column 5, lines 44-46) and “When the user selects a resource, a browser sends an open URL request and cookie to a Protected Web Server. A Protected Web Server is a web server with resources protected by the Runtime Module decrypts information in the cookie and uses it to verify that he user is authorized to access the resource” (Column 6, lines 58-64). The examiner further notes that **Win** teaches **“wherein the assigned metadata requirements specify conditions required for successful execution of the functional module”** as “the runtime module on the protected server receives the login request and intercepts all other request by the client to use a resource” (Abstract), “If the name and password are correct, the Authentication Client Module reads the user’s roles from the Registry server” (Column 6, lines 44-46), and “a personalized menu is an HTML page containing a list of authorized Resources” (Column 6, lines 13-14). The examiner further notes that **Win** teaches **“wherein at least one condition defines at least one user role required for successful; execution of the functional module”** as “the runtime module on the protected server receives the login request and intercepts all other request by the client

to use a resource" (Abstract), "If the name and password are correct, the Authentication Client Module reads the user's roles from the Registry server" (Column 6, lines 44-46), and "a personalized menu is an HTML page containing a list of authorized Resources" (Column 6, lines 13-14). The examiner further notes that **Win** teaches **"obtaining a list of functional modules that are accessible from within the application used during the query session"** as "When the user selects a resource, a browser sends an open URL request and cookie to a Protected Web Server. A Protected Web Server is a web server with resources protected by the Runtime Module decrypts information in the cookie and uses it to verify that he user is authorized to access the resource" (Column 6, lines 58-64). The examiner further notes that **Win** teaches **"identifying a limited subset of the functional modules in the list that will successfully execute, by comparing the collected runtime metadata with the assigned metadata requirements"** as "a Personalized Menu is an HTML page containing a list of authorized resources. The Personalized Menus displays only Resources to which the User has access" (Column 6, lines 12-15) and "When the user selects a resource, a browser sends an open URL request and cookie to a Protected Web Server. A Protected Web Server is a web server with resources protected by the Runtime Module decrypts information in the cookie and uses it to verify that he user is authorized to access the resource" (Column 6, lines 58-64). The examiner further notes that **Win** teaches **"providing an interface presenting the user with the identified limited subset of functional modules that will successfully execute"** as "a Personalized Menu is an HTML page containing a list of authorized resources. The Personalized Menus displays only Resources to which the User has access" (Column 6, lines 12-15).

Win does not explicitly teach:

- D) collecting runtime metadata relating to one or more result fields in a query statement;
- E) wherein the one or more result fields specify one or more data fields for which data is requested to be returned upon execution of the query statement;
- F) wherein the runtime metadata is collected after composition of the query statement.

Faybishenko, however, teaches **“collecting runtime metadata relating to one or more result fields of a query”** as “In one embodiment a QRP adapter may monitor or log queries, results, number of hits, searches, results, etc. or generally the information passing through the QRP adapter. In one embodiment, a user interface may be provided through which providers may view the results of searches and hits performed by consumers--e.g. how many searches resulted in their entry being returned, how many users clicked through, etc. In one embodiment, a user interface may be provided through which providers may monitor and/or control the number of queries sent to them and also to throttle traffic (e.g. turn it off) if necessary. In some embodiments, a QRP interface may be able to access a registration file, for example to read at least part of the registration document or to write to replace or to add to at least part of the registration document” (Paragraph 112), **“wherein the metadata is collected after composition of the query”** as “In one embodiment a QRP adapter may monitor or log queries, results, number of hits, searches, results, etc. or generally the information passing through the QRP adapter. In one embodiment, a user interface may be provided through which providers may view the results of searches and hits performed by consumers--e.g. how many searches resulted in their entry being returned, how many users clicked through, etc. In one embodiment, a user interface may be provided through which providers may monitor and/or control the number of queries sent to them and also to throttle traffic (e.g. turn it off) if necessary. In some embodiments, a QRP interface may be able to access a registration file, for example to read at least part of the registration document or to write to replace or to add to at least part of the registration document” (Paragraph 112), and **“wherein the runtime metadata is collected after composition of the query statement”** as “In one embodiment a QRP adapter may monitor or log queries, results, number of hits, searches, results, etc. or generally the information passing through the QRP adapter. In one embodiment, a user interface may be provided through which providers may view the results of searches and hits performed by consumers--e.g. how many searches resulted in their entry being returned, how many users clicked through, etc. In one embodiment, a user interface may be provided through which providers may monitor

and/or control the number of queries sent to them and also to throttle traffic (e.g. turn it off) if necessary. In some embodiments, a QRP interface may be able to access a registration file, for example to read at least part of the registration document or to write to replace or to add to at least part of the registration document" (Paragraph 112).

The examiner further notes that logging an entire query teaches the claimed result fields because the result fields are encompassed within that query.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching **Faybishenko's** would have allowed **Win's** to provide a method to allow for more control over content provided over the internet to providers, as noted by **Faybishenko** (Paragraph 6).

Win and Faybishenko do not explicitly teach:

G) wherein the runtime metadata is collected before the query statement is submitted for execution.

Chang, however, teaches **"wherein the runtime metadata is collected before the query statement is submitted for execution"** as "The logical operations of the executable begin at signal operation 202 where the CPU 104 awaits a user activity signal that is provided from the logical operations of FIG. 3. The user activity signal specifies when a particular key on the keyboard has been pressed by the user and when a particular mouse button has been clicked by the user. Query operation 204 detects whether the signal has been received. If not, then signal operation 202 continues to await the user activity signal. If the signal of user activity has been received, then query operation 206 detects whether the signal specifies a mouse click as opposed to a keyboard type. When a keyboard type is detected instead of a mouse click, then record operation 208 records the keyboard type as a user-driven event. As discussed, recording the user-driven event may involve one or more techniques, such as displaying a textual description

of the keyboard type on the display screen, saving a description to an electronic file, and/or printing the textual description via a printer. Each character that is typed may be placed on the same line of the textual description as shown in the screenshots until the typed key is an <Enter> key or unless there is a control key such as <Backspace> or <Ctrl>+"A". After each keystroke is recorded, operational flow returns to signal operation 202" (Column 5, lines 4-26).

The examiner notes that **Chang** records, saves, and displays all concurrent typed keyboard strokes by a user. Because **Chang** records the typed keyboard strokes as a user is typing them, then, as a result, the typed keywords are saved before a "submission of execution".

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching **Chang's** would have allowed **Win's** and **Faybishenko's** to provide a method that adequately records user events, as noted by **Chang** (Column 1, lines 49-50).

Regarding claim 14, **Win** further teaches a method comprising:

A) wherein obtaining metadata associated with the functional module comprises examining a signature validation (Column 6, lines 1-3, Column 14, lines 34-43).

The examiner notes that **Win** teaches "**wherein obtaining metadata associated with the functional module comprises examining a signature validation**" as "users may log in either with a digital certificate or by opening a login page URL with a web browser and entering a name and password" (Column 6, lines 1-3).

Regarding claim 15, **Win** further teaches a method comprising:

A) wherein the metadata associated with at least one of the functional modules comprises at least one of: one or more input parameters required for successful execution of the functional module, one or more output parameters required for

successful execution of the functional module, and a security credential required to execute the functional module (Abstract, Column 6, lines 10-16, lines 41-65).

The examiner notes that **Win** teaches **“wherein the metadata associated with at least one of the functional modules comprises at least one of one or more input parameters required for successful execution of the functional module; one or more output parameters required for successful execution of the functional module; and a credential of a user authorized to execute the functional module”** as “The Authentication Client Module and Access Menu Module authenticates a user by verifying the name and password with the Registry Server 108” (Column 6, lines 42-44).

Regarding claim 16, **Win** does not explicitly teach a method comprising:

A) wherein at least one of the functional modules analyzes query results.

Faybishenko, however, teaches **“wherein at least one of the functional modules analyzes query results”** as “In one embodiment a QRP adapter may monitor or log queries, results, number of hits, searches, results, etc. or generally the information passing through the QRP adapter. In one embodiment, a user interface may be provided through which providers may view the results of searches and hits performed by consumers--e.g. how many searches resulted in their entry being returned, how many users clicked through, etc. In one embodiment, a user interface may be provided through which providers may monitor and/or control the number of queries sent to them and also to throttle traffic (e.g. turn it off) if necessary. In some embodiments, a QRP interface may be able to access a registration file, for example to read at least part of the registration document or to write to replace or to add to at least part of the registration document” (Paragraph 112).

The examiner further notes that storing the click history of users teaches the claimed analyzing of query results.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching **Faybishenko's** would have allowed **Win's** to provide a method to allow for more control

over content provided over the internet to providers, as noted by **Faybisenko** (Paragraph 6).

Regarding claim 20, **Win** teaches computer readable storage medium comprising:

- A) assigning metadata requirements to functional modules that operate on data stored in, or functional modules that generate results that are stored in, a database (Abstract, Column 5, lines 44-46, Column 6, lines 10-16, lines 41-65);
- B) wherein the assigned metadata requirements specify conditions required for successful execution of the functional module (Abstract, Column 5, lines 44-46, Column 6, lines 10-16, lines 41-65);
- C) wherein at least one condition defines at least one user role required for successful execution of the functional module (Abstract, Column 5, lines 44-46, Column 6, lines 10-16, lines 41-65);
- H) obtaining a list of functional modules accessible from within the application (Abstract, Column 6, lines 10-16, lines 41-65);
- I) identifying a limited subset of the functional modules that will successfully execute, by comparing the collected runtime metadata with the assigned metadata requirements (Abstract, Column 6, lines 10-16, lines 41-65); and
- J) providing an interface presenting the user with the identified limited subset of functional modules that will successfully execute (Abstract, Column 6, lines 10-16, lines 41-65).

The examiner notes that **Win** teaches **“assigning metadata requirements to functional modules that operate on data stored in, or functional modules that generate results that are stored in, a database”** as “Roles determine what resources a User can access. Further, each role may require a set of information that is available in resources” (Column 5, lines 44-46) and “When the user selects a resource, a browser sends an open URL request and cookie to a Protected Web Server. A Protected Web Server is a web server with resources protected by the Runtime Module decrypts information in the cookie and uses it to verify that he user is authorized to

access the resource" (Column 6, lines 58-64). The examiner further notes that **Win** teaches **"wherein the assigned metadata requirements specify conditions required for successful execution of the functional module"** as "the runtime module on the protected server receives the login request and intercepts all other request by the client to use a resource" (Abstract), "If the name and password are correct, the Authentication Client Module reads the user's roles from the Registry server" (Column 6, lines 44-46), and "a personalized menu is an HTML page containing a list of authorized Resources" (Column 6, lines 13-14). The examiner further notes that **Win** teaches **"wherein at least one condition defines at least one user role required for successful; execution of the functional module"** as "the runtime module on the protected server receives the login request and intercepts all other request by the client to use a resource" (Abstract), "If the name and password are correct, the Authentication Client Module reads the user's roles from the Registry server" (Column 6, lines 44-46), and "a personalized menu is an HTML page containing a list of authorized Resources" (Column 6, lines 13-14). The examiner further notes that **Win** teaches **"obtaining a list of functional modules accessible from within the application"** as "a list of authorized resources" (Column 6, lines 13-14) and "When the user selects a resource, a browser sends an open URL request and cookie to a Protected Web Server. A Protected Web Server is a web server with resources protected by the Runtime Module decrypts information in the cookie and uses it to verify that he user is authorized to access the resource" (Column 6, lines 58-64). The examiner further notes that **Win** teaches **"identifying a limited subset of the functional modules that will successfully execute, by comparing the collected runtime metadata with the assigned metadata requirements"** as "a Personalized Menu is an HTML page containing a list of authorized resources. The Personalized Menus displays only Resources to which the User has access" (Column 6, lines 12-15) and "When the user selects a resource, a browser sends an open URL request and cookie to a Protected Web Server. A Protected Web Server is a web server with resources protected by the Runtime Module decrypts information in the cookie and uses it to verify that he user is authorized to access the resource" (Column 6, lines 58-64). The examiner further notes

that **Win** teaches “**providing an interface presenting the user with the identified limited subset of functional modules that will successfully execute**” as “a Personalized Menu is an HTML page containing a list of authorized resources. The Personalized Menu displays only Resources to which the User has access” (Column 6, lines 12-15).

Win does not explicitly teach:

- D) collecting runtime metadata relating to one or more result fields in a query statement;
- E) wherein the one or more result fields specify one or more data fields for which data is requested to be returned upon execution of the query statement;
- F) wherein the runtime metadata is collected after composition of the query statement.

Faybishenko, however, teaches “**collecting runtime metadata relating to one or more result fields in a query statement**” as “In one embodiment a QRP adapter may monitor or log queries, results, number of hits, searches, results, etc. or generally the information passing through the QRP adapter. In one embodiment, a user interface may be provided through which providers may view the results of searches and hits performed by consumers--e.g. how many searches resulted in their entry being returned, how many users clicked through, etc. In one embodiment, a user interface may be provided through which providers may monitor and/or control the number of queries sent to them and also to throttle traffic (e.g. turn it off) if necessary. In some embodiments, a QRP interface may be able to access a registration file, for example to read at least part of the registration document or to write to replace or to add to at least part of the registration document” (Paragraph 112), “**wherein the one or more result fields specify one or more data fields for which data is requested to be returned upon execution of the query statement**” as “In one embodiment a QRP adapter may monitor or log queries, results, number of hits, searches, results, etc. or generally the information passing through the QRP adapter. In one embodiment, a user interface may be provided through which providers may view the results of searches and hits performed by consumers--e.g. how many searches resulted in their entry being returned, how many users clicked through, etc. In one embodiment, a user interface

may be provided through which providers may monitor and/or control the number of queries sent to them and also to throttle traffic (e.g. turn it off) if necessary. In some embodiments, a QRP interface may be able to access a registration file, for example to read at least part of the registration document or to write to replace or to add to at least part of the registration document" (Paragraph 112), and **"wherein the runtime metadata is collected after composition of the query statement"** as "In one embodiment a QRP adapter may monitor or log queries, results, number of hits, searches, results, etc. or generally the information passing through the QRP adapter. In one embodiment, a user interface may be provided through which providers may view the results of searches and hits performed by consumers--e.g. how many searches resulted in their entry being returned, how many users clicked through, etc. In one embodiment, a user interface may be provided through which providers may monitor and/or control the number of queries sent to them and also to throttle traffic (e.g. turn it off) if necessary. In some embodiments, a QRP interface may be able to access a registration file, for example to read at least part of the registration document or to write to replace or to add to at least part of the registration document" (Paragraph 112).

The examiner further notes that logging an entire query teaches the claimed result fields because the result fields are encompassed within that query.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching **Faybishenko's** would have allowed **Win's** to provide a method to allow for more control over content provided over the internet to providers, as noted by **Faybishenko** (Paragraph 6).

Win and Faybishenko do not explicitly teach:

G) wherein the runtime metadata is collected before the query statement is submitted for execution.

Chang, however, teaches **"wherein the runtime metadata is collected before the query statement is submitted for execution"** as "The logical operations of the executable begin at signal operation 202 where the CPU 104 awaits a user activity signal that is provided from the

logical operations of FIG. 3. The user activity signal specifies when a particular key on the keyboard has been pressed by the user and when a particular mouse button has been clicked by the user. Query operation 204 detects whether the signal has been received. If not, then signal operation 202 continues to await the user activity signal. If the signal of user activity has been received, then query operation 206 detects whether the signal specifies a mouse click as opposed to a keyboard type. When a keyboard type is detected instead of a mouse click, then record operation 208 records the keyboard type as a user-driven event. As discussed, recording the user-driven event may involve one or more techniques, such as displaying a textual description of the keyboard type on the display screen, saving a description to an electronic file, and/or printing the textual description via a printer. Each character that is typed may be placed on the same line of the textual description as shown in the screenshots until the typed key is an <Enter> key or unless there is a control key such as <Backspace> or <Ctrl>+"A". After each keystroke is recorded, operational flow returns to signal operation 202" (Column 5, lines 4-26).

The examiner notes that **Chang** records, saves, and displays all concurrent typed keyboard strokes by a user. Because **Chang** records the typed keyboard strokes as a user is typing them, then, as a result, the typed keywords are saved before a "submission of execution".

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching **Chang's** would have allowed **Win's** and **Faybishenko's** to provide a method that adequately records user events, as noted by **Chang** (Column 1, lines 49-50).

Regarding claim 21, **Win** does not explicitly teach a computer readable storage medium comprising:

A) wherein the application is a query building application.

Faybishenko, however, teaches “**wherein the application is a query building application**” as “In one embodiment a QRP adapter may monitor or log queries, results, number of hits, searches, results, etc. or generally the information passing through the QRP adapter. In one embodiment, a user interface may be provided through which providers may view the results of searches and hits performed by consumers--e.g. how many searches resulted in their entry being returned, how many users clicked through, etc. In one embodiment, a user interface may be provided through which providers may monitor and/or control the number of queries sent to them and also to throttle traffic (e.g. turn it off) if necessary. In some embodiments, a QRP interface may be able to access a registration file, for example to read at least part of the registration document or to write to replace or to add to at least part of the registration document” (Paragraph 112)

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching **Faybishenko’s** would have allowed **Win’s** to provide a method to allow for more control over content provided over the internet to providers, as noted by **Faybishenko** (Paragraph 6).

Regarding claim 27, **Win** teaches a data processing system comprising:

A) a data repository (Column 5, lines 13-15);

B) a plurality of functional modules, each having associated metadata requirements that specify conditions required for successful execution of the functional modules (Abstract, Column 6, lines 10-16, lines 41-65);

C) wherein at least one condition defines at least one user role required for successful execution of the functional modules (Abstract, Column 5, lines 44-46, Column 6, lines 10-16, lines 41-65);

D) an application from which the functional modules are accessible (Abstract, Column 6, lines 10-16, lines 41-65);

G) present to a user a limited subset of the functional modules that will successfully execute, as determined by the application based on the collected runtime metadata and the metadata requirements associated with the functional modules (Abstract, Column 5, lines 66-67-Column 6, lines 1-16).

The examiner notes that Win teaches **“a data repository”** as “The system 2 enables organizations to register information sources or Resources and register Users of the information in a central repository” (Column 5, lines 13-15). The examiner further notes that Win teaches **“a plurality of functional modules, each having associated metadata requirements that specify conditions required for successful execution of the functional modules”** as “a list of authorized resources” (Column 6, lines 13-14). The examiner further notes that Win teaches **“an application from which the functional modules are accessible”** as “a personalized menu is an HTML page containing a list of authorized Resources” (Column 6, lines 13-14). The examiner further notes that Win teaches **“wherein at least one condition defines at least one user role required for successful execution of the functional modules”** as “the runtime module on the protected server receives the login request and intercepts all other request by the client to use a resource” (Abstract), “If the name and password are correct, the Authentication Client Module reads the user’s roles from the Registry server” (Column 6, lines 44-46), and “a personalized menu is an HTML page containing a list of authorized Resources” (Column 6, lines 13-14). The examiner further notes that Win teaches **“present to a user a limited subset of the functional modules that will successfully execute, as determined by the application based on the collected runtime metadata and the metadata requirements associated with the functional modules”** as “the runtime module on the protected server receives the login request and intercepts all other request by the client to use a resource” (Abstract), “If the name and password are correct, the Authentication Client Module reads the user’s roles from the Registry server” (Column 6, lines 44-46), and “a personalized menu is an HTML page containing a list of authorized Resources” (Column 6, lines 13-14).

Win does not explicitly teach:

F) wherein the one or more result fields specify one or more data fields for which data is requested to be returned upon execution of the query statement;

Faybishenko, however, teaches “**wherein the one or more result fields specify one or more data fields for which data is requested to be returned upon execution of the query statement**” as “In one embodiment a QRP adapter may monitor or log queries, results, number of hits, searches, results, etc. or generally the information passing through the QRP adapter. In one embodiment, a user interface may be provided through which providers may view the results of searches and hits performed by consumers--e.g. how many searches resulted in their entry being returned, how many users clicked through, etc. In one embodiment, a user interface may be provided through which providers may monitor and/or control the number of queries sent to them and also to throttle traffic (e.g. turn it off) if necessary. In some embodiments, a QRP interface may be able to access a registration file, for example to read at least part of the registration document or to write to replace or to add to at least part of the registration document” (Paragraph 112).

The examiner further notes that logging an entire query teaches the claimed result fields because the result fields are encompassed within that query.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching **Faybishenko's** would have allowed **Win's** to provide a method to allow for more control over content provided over the internet to providers, as noted by **Faybishenko** (Paragraph 6).

Win and **Faybishenko** do not explicitly teach:

E) wherein the application is configured to: after composition of a query statement, but before the query statement is submitted for execution, collect runtime metadata related to one or more result fields in the query statement.

Chang, however, teaches “**wherein the application is configured to: after composition of a query statement, but before the query statement is submitted for execution, collect runtime metadata related to one or more result fields in the**

query statement" as "The logical operations of the executable begin at signal operation 202 where the CPU 104 awaits a user activity signal that is provided from the logical operations of FIG. 3. The user activity signal specifies when a particular key on the keyboard has been pressed by the user and when a particular mouse button has been clicked by the user. Query operation 204 detects whether the signal has been received. If not, then signal operation 202 continues to await the user activity signal. If the signal of user activity has been received, then query operation 206 detects whether the signal specifies a mouse click as opposed to a keyboard type. When a keyboard type is detected instead of a mouse click, then record operation 208 records the keyboard type as a user-driven event. As discussed, recording the user-driven event may involve one or more techniques, such as displaying a textual description of the keyboard type on the display screen, saving a description to an electronic file, and/or printing the textual description via a printer. Each character that is typed may be placed on the same line of the textual description as shown in the screenshots until the typed key is an <Enter> key or unless there is a control key such as <Backspace> or <Ctrl>+"A". After each keystroke is recorded, operational flow returns to signal operation 202" (Column 5, lines 4-26).

The examiner notes that **Chang** records, saves, and displays all concurrent typed keyboard strokes by a user. Because **Chang** records the typed keyboard strokes as a user is typing them, then, as a result, the typed keywords are saved before a "submission of execution".

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching

Chang's would have allowed **Win's** and **Faybishenko's** to provide a method that adequately records user events, as noted by **Chang** (Column 1, lines 49-50).

Regarding claim 28, **Win** does not explicitly teach a data processing system comprising:

A) wherein the data repository comprises XML data structures used to store runtime metadata.

Faybishenko, however, teaches "**wherein the data repository comprises XML data structures used to store runtime metadata**" as "In some embodiments, users and end applications (consumers 140) may present queries to a distributed information discovery network as arbitrary XML. Schema selection may be performed by HTTP header specification, in some embodiments. In one embodiment, queries presented by consumers 140 may adhere to specific queriespaces. In some embodiments, queries may be routed to the appropriate provider 120 by sending requests (e.g. XML requests) over HTTP" (Paragraph 54).

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching **Faybishenko's** would have allowed **Win's** to provide a method to allow for more control over content provided over the internet to providers, as noted by **Faybishenko** (Paragraph 6).

Regarding claim 29, **Win** further teaches a data processing system comprising:

A) wherein the data repository comprises relational database tables used to store runtime metadata (Column 5, lines 13-15, Column 7, lines 1-6).

The examiner notes that **Win** teaches "**wherein the data repository comprises relational database tables used to store runtime metadata**" as "The Registry Repository is structured as a database. For example, the Registry Repository may be an SQL Server relational database management system, the Oracle 7® database, etc." (Column 7, lines 1-6). The examiner further notes that it is common knowledge that relational databases store data in tables.

Regarding claim 30, **Win** does not explicitly teach a method comprising:

A) wherein the runtime metadata relating to one or more result fields in the query statement comprises one or more of: a result field name; and a data type.

Faybishenko, however, teaches “**wherein the runtime metadata relating to one or more result fields in the query statement comprises one or more of: a result field name; and a data type**” as “In one embodiment a QRP adapter may monitor or log queries, results, number of hits, searches, results, etc. or generally the information passing through the QRP adapter. In one embodiment, a user interface may be provided through which providers may view the results of searches and hits performed by consumers--e.g. how many searches resulted in their entry being returned, how many users clicked through, etc. In one embodiment, a user interface may be provided through which providers may monitor and/or control the number of queries sent to them and also to throttle traffic (e.g. turn it off) if necessary. In some embodiments, a QRP interface may be able to access a registration file, for example to read at least part of the registration document or to write to replace or to add to at least part of the registration document” (Paragraph 112).

The examiner further notes that logging an entire query teaches the claimed result fields because the result fields are encompassed within that query.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching **Faybishenko's** would have allowed **Win's** to provide a method to allow for more control over content provided over the internet to providers, as noted by **Faybishenko** (Paragraph 6).

Regarding claim 31, **Win** does not explicitly teach a computer readable storage medium comprising:

A) wherein the runtime metadata relating to one or more result fields in the query statement comprises one or more of: a result field name; and a data type for the result field.

Faybishenko, however, teaches “**wherein the runtime metadata relating to one or more result fields in the query statement comprises one or more of: a result field name; and a data type for the result field**” as “In one embodiment a QRP adapter may monitor or log queries, results, number of hits, searches, results, etc. or generally the information passing through the QRP adapter. In one embodiment, a user interface may be provided through which providers may view the results of searches and hits performed by consumers—e.g. how many searches resulted in their entry being returned, how many users clicked through, etc. In one embodiment, a user interface may be provided through which providers may monitor and/or control the number of queries sent to them and also to throttle traffic (e.g. turn it off) if necessary. In some embodiments, a QRP interface may be able to access a registration file, for example to read at least part of the registration document or to write to replace or to add to at least part of the registration document” (Paragraph 112).

The examiner further notes that logging an entire query teaches the claimed result fields because the result fields are encompassed within that query.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching **Faybishenko’s** would have allowed **Win’s** to provide a method to allow for more control over content provided over the internet to providers, as noted by **Faybishenko** (Paragraph 6).

Regarding claim 32, **Win** does not explicitly teach a data processing system comprising:

A) wherein the runtime metadata relating to one or more result fields in the query statement comprises one or more of: a result field name; and a data type for the result field.

Faybishenko, however, teaches “**wherein the runtime metadata relating to one or more result fields in the query statement comprises one or more of: a result field name; and a data type for the result field**” as “In one embodiment a QRP adapter may monitor or log queries, results, number of hits, searches, results, etc. or

generally the information passing through the QRP adapter. In one embodiment, a user interface may be provided through which providers may view the results of searches and hits performed by consumers--e.g. how many searches resulted in their entry being returned, how many users clicked through, etc. In one embodiment, a user interface may be provided through which providers may monitor and/or control the number of queries sent to them and also to throttle traffic (e.g. turn it off) if necessary. In some embodiments, a QRP interface may be able to access a registration file, for example to read at least part of the registration document or to write to replace or to add to at least part of the registration document" (Paragraph 112).

The examiner further notes that logging an entire query teaches the claimed result fields because the result fields are encompassed within that query.

It would have been obvious to one of ordinary skill in the art at the time the invention was made to combine the teachings of the cited references because teaching **Faybishenko's** would have allowed **Win's** to provide a method to allow for more control over content provided over the internet to providers, as noted by **Faybishenko** (Paragraph 6).

Response to Arguments

6. Applicant's arguments with respect to claims 10, 14-16, 20-21, and 27-32 have been considered but are moot in view of the new ground(s) of rejection.

Conclusion

7. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure.

U.S. PGPUB 2002/0083075 issued to **Brummel et al.** on 27 June 2002. The subject matter disclosed therein is pertinent to that of claims 10, 14-16, 20-21, and 27-32 (e.g., methods to use plug-ins in an interface for role-based users).

U.S. Patent 6,757,898 issued to **Ilse et al.** on 29 June 2004. The subject matter disclosed therein is pertinent to that of claims 10, 14-16, 20-21, and 27-32 (e.g., methods to use plug-ins in an interface for role-based users).

U.S. PGPUB 2003/0140043 issued to **Hotchkiss et al.** on 24 July 2003. The subject matter disclosed therein is pertinent to that of claims 10, 14-16, 20-21, and 27-32 (e.g., methods to use plug-ins in an interface for role-based users).

U.S. PGPUB 2004/0249674 issued to **Eisenberg** on 09 December 2004. The subject matter disclosed therein is pertinent to that of claims 10, 14-16, 20-21, and 27-32 (e.g., methods to use plug-ins in an interface for role-based users).

U.S. PGPUB 2003/0229623 issued to **Chang et al.** on 11 December 2003. The subject matter disclosed therein is pertinent to that of claims 10, 14-16, 20-21, and 27-32 (e.g., methods to use plug-ins in an interface for role-based users).

U.S. PGPUB 2002/0091836 issued to **Moetelli** on 11 July 2002. The subject matter disclosed therein is pertinent to that of claims 10, 14-16, 20-21, and 27-32 (e.g., methods to use plug-ins in an interface for role-based users).

U.S. Patent 6,430,556 issued to **Goldberg et al.** on 06 August 2002. The subject matter disclosed therein is pertinent to that of claims 10, 14-16, 20-21, and 27-32 (e.g., methods to use plug-ins in an interface for role-based users).

U.S. Patent 7,027,975 issued to **Pazandak et al.** on 11 April 2006. The subject matter disclosed therein is pertinent to that of claims 10, 14-16, 20-21, and 27-32 (e.g., methods to use plug-ins in an interface for role-based users).

U.S. PGPUB 2005/0080656 issued to **Crow et al.** on 14 April 2005. The subject matter disclosed therein is pertinent to that of claims 10, 14-16, 20-21, and 27-32 (e.g., methods to use plug-ins in an interface for role-based users).

U.S. PGPUB 2002/0147724 issued to **Fries et al.** on 10 October 2002. The subject matter disclosed therein is pertinent to that of claims 10, 14-16, 20-21, and 27-32 (e.g., methods to use plug-ins in an interface for role-based users).

U.S. Patent 6,938,035 issued to **Driesch et al.** on 30 August 2005. The subject matter disclosed therein is pertinent to that of claims 10, 14-16, 20-21, and 27-32 (e.g., methods to use plug-ins in an interface for role-based users).

U.S. PGPUB 2003/0065648 issued to **Driesch et al.** on 03 April 2003. The subject matter disclosed therein is pertinent to that of claims 10, 14-16, 20-21, and 27-32 (e.g., methods to use plug-ins in an interface for role-based users).

8. Applicant's amendment necessitated the new ground(s) of rejection presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**. See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE MONTHS from the mailing date of this action. In the event a first reply is filed within TWO MONTHS of the mailing date of this final action and the advisory action is not mailed until after the end of the THREE-MONTH shortened statutory period, then the shortened statutory period will expire on the date the advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a) will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

Contact Information

9. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Mahesh Dwivedi whose telephone number is (571) 272-2731. The examiner can normally be reached on Monday to Friday 8:20 am – 4:40 pm.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Tim Vo can be reached (571) 272-3642. The fax number for the organization where this application or proceeding is assigned is (571) 273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see <http://pair-direct.uspto.gov>. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

Mahesh Dwivedi
Patent Examiner
Art Unit 2168

July 29, 2010
/Mahesh H Dwivedi/
Examiner, Art Unit 2168

/Tim T. Vo/
Supervisory Patent Examiner, Art Unit 2168